ICCV
#10075

ICCV
#10075

ICCV 2025 Submission #10075. CONFIDENTIAL REVIEW COPY. DO NOT DISTRIBUTE.

TrackAny3D: Transferring Pretrained 3D Models for Category-unified 3D Point Cloud Tracking

Anonymous ICCV submission

Paper ID 10075

Abstract

3D LiDAR-based single object tracking (SOT) relies on sparse and irregular point clouds, posing challenges from geometric variations in scale, motion patterns, and structural complexity across object categories. Current category-specific approaches achieve good accuracy but are impractical for real-world use, requiring separate models for each category and showing limited generalization. To tackle these issues, we propose TrackAny3D, the first framework to transfer large-scale pretrained 3D models for category-agnostic 3D SOT. We first integrate parameter-efficient adapters to bridge the gap between pretraining and tracking tasks while preserving geometric priors. Then, we introduce a Mixture-of-Geometry-Experts (MoGE) architecture that adaptively activates specialized subnetworks based on distinct geometric characteristics. Additionally, we design a temporal context optimization strategy that incorporates learnable temporal tokens and a dynamic mask weighting module to propagate historical information and mitigate temporal drift. Experiments on three commonly-used benchmarks show that TrackAny3D establishes new state-of-the-art performance on category-agnostic 3D SOT, demonstrating strong generalization and competitiveness. We hope this work will enlighten the community on the importance of unified models and further expand the use of large-scale pretrained models in this field. The source code will be released.

1. Introduction

3D SOT on point clouds [10, 15, 34] is a task of persistently localizing a target in a dynamic 3D scene. This task holds substantial potential for a wide range of applications, such as autonomous driving and mobile robotics. Unlike RGB-based tracking [3, 19, 50], which benefits from rich texture and color cues, 3D LiDAR-based SOT relies exclusively on sparse, irregular point clouds to infer a target's 3D spatial pose. This geometric dependency introduces unique chal-
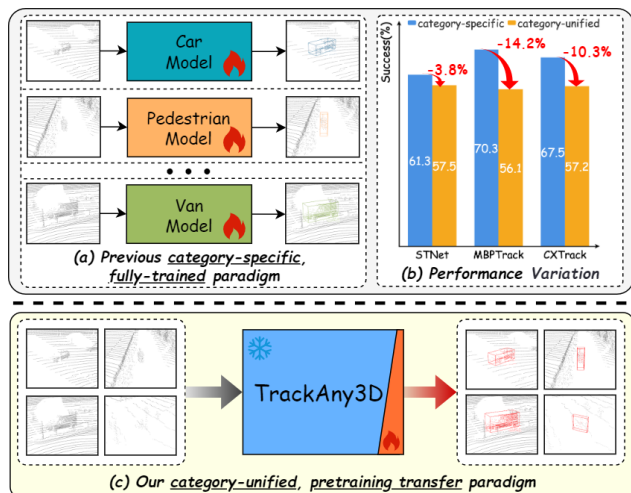

Figure 1. **Comparison between Different Tracking Paradigms.** The previous category-specific, fully trained paradigm (a) employs multiple models, each learned for a specific category. We observed that they face significant performance drops when using cross-category training (on KITTI) (b). In contrast, our category-unified, pretraining transfer paradigm (c) uses a single shared model for all categories, which is efficiently transferred from pretrained 3D models.

lenges: objects from different categories (e.g., cars, pedestrians) exhibit drastic variations in scale, motion patterns, and structural complexity.

To mitigate these challenges, existing methods [24, 26, 29, 41, 48] adopt a category-specific learning paradigm as shown in Fig. 1(a), where plenty of dedicated models are independently trained and tested for each object type. While attaining leading accuracy, this paradigm is impractical for real-world deployment, as it demands prohibitive computational resources to train and store dozens of category-specific networks, and fails to generalize to novel categories, a critical limitation for open-world applications. To validate the inter-class geometric variation challenge, we empirically observe that directly applying existing methods [15, 47, 48] to train unified models on all categories

ICCV
#10075

ICCV
#10075

ICCV 2025 Submission #10075. CONFIDENTIAL REVIEW COPY. DO NOT DISTRIBUTE.

leads to severe performance degradation compared to their category-specific counterparts, as shown in Fig. 1(b). Mo-CUT [28] is the only method that attempts to solve this problem by explicitly encoding distinct attributes associated with different object categories. However, it mainly enforces uniformity through non-learnable constraints, which demands manual hyper-parameter tuning and limits generalization. Based on the above analysis, we are interested in the question: **How can we learn geometry-aware yet category-agnostic representations without introducing manual biases?**

The rise of large-scale pretrained models [21, 31, 36, 40] provides a transformative and promising solution. In the fields of 2D vision [19, 30, 42] and natural language processing (NLP) [5, 23], foundation models pretrained on web-scale data have demonstrated remarkable downstream generalization. This is enabled by parameter-efficient fine-tuning transfer (PEFT) techniques such as prompt tuning [16, 60] or adapter modules [8, 11]. Similarly, we believe that pretrained 3D point cloud models [31, 46, 55] could provide valuable geometric priors for 3D SOT, potentially alleviating the aforementioned geometric disparity challenges to some extent. However, extending this paradigm to 3D SOT remains largely unexplored and primarily faces three fundamental barriers: (i) distribution mismatch, where pretraining datasets (e.g., ShapeNet [2], ScanNet [4]) have limited category diversity and scene complexity compared to real-world tracking scenarios [1, 9]; (ii) persistent gaps, where pretrained models partially alleviate geometric disparities but fail to fully resolve the intrinsic conflict between geometric sensitivity; and (iii) lack of temporal modeling, as pretraining tasks focus on static shape reconstruction [46, 55] or recognition [31, 51], while tracking requires modeling temporal coherence.

To address these challenges, we propose TrackAny3D, the first framework to effectively transfer large-scale pretrained point cloud models for category-agnostic 3D SOT. TrackAny3D follows a novel category-unified, pretraining transfer paradigm, as shown in Fig. 1(c), and consists of three core designs, each addressing one of the three aforementioned problems. Specifically, we integrate a lightweight, two-path adapter into the Transformer layers. One path within the adapter handles feature adaptation, while the other path modulates the intensity of this adaptation. This adapter dynamically aligns pretrained features with 3D SOT tasks while freezing the original pretrained network to preserve geometric priors and enhance learning efficiency. In order to further address the persistent gaps, we introduce Mixture-of-Geometry-Experts (MoGE) which consists of several expert subnetworks, where specialized experts are adaptively activated based on object geometry, resolving conflicts between divergent geometric patterns. Additionally, considering temporal modeling, we propose

temporal context optimization strategies that propagate historical states through learnable temporal tokens and adaptively calibrate input information based on real-time geometric changes via a dynamic mask weighting mechanism. Extensive experiments conducted on KITTI [9], NuScenes [1], and Waymo [39] demonstrate that our Track-Any3D achieves state-of-the-art (SOTA) performance under the category-unified setting, while also showcasing robust generalization capabilities.

Our main contributions can be summarized as: **1)** We propose TrackAny3D, the first method to successfully adapt large-scale 3D pretrained models to open-world 3D SOT without category-specific tuning, by integrating lightweight adapters that enable effective knowledge transfer. **2)** We introduce a mixture-of-geometry-experts architecture where each expert subnetwork learns distinct geometric characteristics to ensure unified yet adaptive processing across diverse object categories. **3)** We design temporal propagation strategies with learnable temporal tokens coupled with a dynamic mask weighting mechanism, which jointly addresses temporal variations and state drift.

2. Related Works

2.1. 3D Single Object Tracking

Since the advent of the pioneering 3D LiDAR-based tracker SC3D [10], this field has witnessed rapid development in recent years [24, 26, 29, 57]. Current 3D SOT methods can be broadly categorized into three types. The first type is Siamese-based methods, which extract features from template and search regions using shared backbones and perform matching via similarity learning. Most existing tracking methods fall into this category [7, 10, 14, 15, 22, 29, 33, 34, 47, 48, 53, 56]. The second type is motion-centric methods [20, 29, 49, 57], which leverage the target's prior state from the previous frame to infer relative motion offsets in the current frame. Another type is unified modeling methods [26, 41], which jointly encode the template and search regions within a single Transformer architecture, unifying feature extraction and matching into an end-to-end framework. Our method adopts this approach to simplify the overall framework and maintain consistency with the input format of pretraining tasks.

Despite the progress made, most existing 3D SOT methods remain category-specific, meaning they are typically trained and fine-tuned for particular object categories, which is undesirable in real-world applications. Only recently has MoCUT [28] focused on this issue. However, MoCUT's solution often relies on manually designed rules and hyperparameter tuning. In this work, we address this problem by adapting the pretraining transfer paradigm and utilizing an adaptive network to overcome this limitation.

ICCV
#10075

ICCV 2025 Submission #10075. CONFIDENTIAL REVIEW COPY. DO NOT DISTRIBUTE.

ICCV
#10075

2.2. PEFT in Point Clouds

The rapid advancement of large-scale pretrained models, exemplified by CLIP[36], LLaVA[21], and Llama[40], has driven the progress in core areas such as 2D vision and NLP through their superior representation learning capabilities. In the point cloud modality, several powerful large-scale models have also emerged, including PointCLIP[55], Point-MAE[31], Point-BERT[51], CLIP2Point[13], and RECON[35], achieving excellent results in tasks like classification and generation. The advancement of large-scale models has also propelled the progress of downstream tasks. Apart from full fine-tuning, PEFT techniques [8, 11, 12, 16, 17, 54, 60] have been widely adopted, which allow models to be adapted efficiently with minimal resource consumption. In the point cloud field, research on PEFT remains relatively limited. Existing methods include IDAT[52], which combines DGCNN[43] with instance-aware prompt extraction for geometric alignment; and DAPT[61], which integrates dynamic adapters with prompt tuning through internal prompts to adaptively capture domain variations.

However, in 3D SOT of the point cloud, the integration of large pretrained models with PEFT remains largely unexplored. To the best of our knowledge, the only related work is MemDisst [45], which uses 3D pretraining for initialization. Nevertheless, it lacks category-unified tracking, requires learning the entire network, and relies on distilling knowledge from a 2D pretrained tracker [50] to ensure performance rather than leveraging PEFT techniques. This limits its efficiency and ability to fully exploit pretrained 3D knowledge. In contrast, this paper aims to fully utilize 3D pretrained models in combination with PEFT, addressing the limitations of category-specific approaches and significantly enhancing model generalizability and unification capabilities.

3. Methodology

3.1. Overview

Task Definition. In a dynamic 3D scene, 3D LiDAR-based SOT aims to localize a target within a sampled search region $\mathcal{P}^s = \{p_i^s\}_{i=1}^{N_s}$ in each frame, given a sampled template region $\mathcal{P}^t = \{p_i^t\}_{i=1}^{N_t}$, which corresponds to the target region from the initial or historical frames. The target's location is defined by a 3D bounding box (BBox) $B \in \mathbb{R}^7$, typically parameterized by the target's center coordinates (x, y, z), orientation angle θ (the yaw angle around the vertical axis), and dimensions (width w, length l, height h). By applying translation and rotation to the template BBox B_t, the BBox B_s for the current search frame can be calculated. Given that the size of the target remains consistent across all frames, only four parameters (x, y, z, θ) need to be predicted for B_t.

Framework Overview. The overall framework is illustrated in Fig. 2(a). For a given search region $\mathcal{P}^s$ and its corresponding template region $\mathcal{P}^t$, we first use a patch embedding layer to extract local information from each region and embed them into tokens. In addition to the point clouds themselves, we also introduce masks $\mathcal{M}^t$ and $\mathcal{M}^s$ [48, 57]. These masks are first processed through our dynamic mask weighting module (Sec. 3.4) and then added element-wise to their corresponding point cloud tokens. Furthermore, we incorporate a learnable temporal token $\mathcal{T}_0$ for temporal information propagation (Sec. 3.4). The template and search tokens, along with the temporal token, are concatenated to form the input $\mathbf{F}_0$ to the encoder. The input $\mathbf{F}_0$ is then passed through an encoder transferred from a pretrained model, which consists of multiple frozen Transformer blocks augmented with our proposed parameter-efficient adapters (Sec. 3.2) and geometry-aware experts architecture (Sec. 3.3). Finally, the output search tokens from the encoder are fed into a localization head [48] to predict bounding boxes for the input search region and the output encoded temporal token is retained to provide contextual cues for the next search frame.

3.2. Efficient Pretrained Model Transfer

TrackAny3D utilizes RECON [35] as the pretrained model, which is a powerful 3D representation learning framework combining the advantages of generative masked modeling and contrastive modeling. Here, we provide a brief introduction to its architecture. Formally, given an input point cloud $\mathcal{P} \in \mathbb{R}^{N \times 3}$ with N points, RECON employs a lightweight PointNet [32] as the patch embedding layer to encode it into input embeddings $\mathbf{F}_0 \in \mathbb{R}^{N \times d}$, where d represents the embedded feature dimension. Next, similarly to ViT [6], the encoder of RECON consists of L Transformer layers, which encode the input tokens $\mathbf{F}_0$. Specifically, each Transformer layer mainly consists of a standard multi-head self-attention (MHSA) block, layer normalizations (LN), and a feed-forward network (FFN). Formally, for the i-th layer:

$$\hat{\mathbf{F}}_{i-1} = \text{MHSA}(\text{LN}(\mathbf{F}_{i-1})) + \mathbf{F}_{i-1}, \qquad (1)$$

$$\mathbf{F}_i = \text{FFN}(\text{LN}(\hat{\mathbf{F}}_{i-1})) + \hat{\mathbf{F}}_{i-1}, \qquad (2)$$

To maintain consistency with the input of the pretrained model, we adopt a unified modeling framework that concatenates a learnable temporal token, the embedded point tokens of the template frame and the search frame to formulate $\mathbf{F}_0 \in \mathbb{R}^{(1+N_t+N_s) \times d}$, while performing feature extraction and matching through unified Transformer blocks. In fact, the most straightforward method to transfer the pretrained model is to fully fine-tune it; however, we found this can lead to suboptimal performance (Table 4) and resource-intensive training. This is because when the model overwrites the knowledge learned during pretraining, it can result in a degradation of its original capabilities [30, 42].
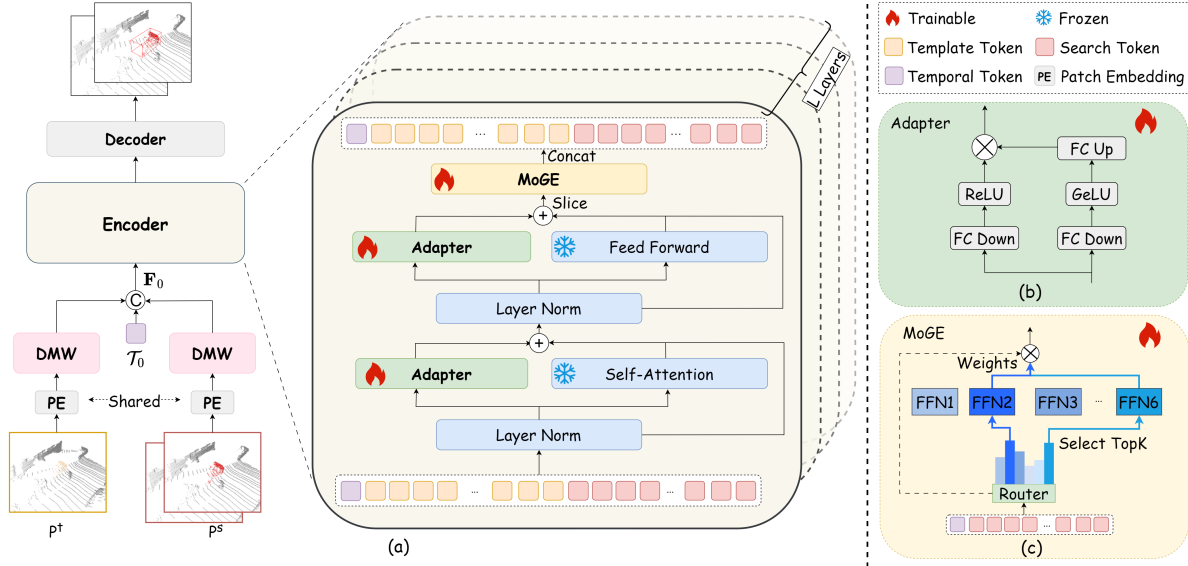
ICCV
#10075

ICCV
#10075

ICCV 2025 Submission #10075. CONFIDENTIAL REVIEW COPY. DO NOT DISTRIBUTE.

Figure 2. **(a) An overview of our proposed TrackAny3D architecture.** Our approach introduces a pretrained encoder, where we freeze the parameters of each Transformer layer. We then adapt it using a lightweight, two-path adapter and incorporate a mixture-of-geometry-expert (MoGE) module for further geometric modeling. Additionally, we introduce a learnable temporal token and a dynamic mask weight (DMW) mechanism to propagate and rectify temporal information. This ultimately results in an effective 3D tracker that shares a single model across all categories. **(b) Details of the adapter.** The adapter comprises two pathes: one dedicated to performing the adaptation and the other to regulating the intensity of this adaptation. **(c) Details of MoGE.** It employs a Router to filter and select different geometry expert sub-networks (FFNs).

Therefore, we explore the PEFT approach, which enables the model to adapt to new tasks with a small number of learnable parameters [61] while preserving the pretrained knowledge by keeping its core parameters frozen.

Specifically, as shown in Fig. 2(b), our adapter module includes two paths: an adaptation path and a gated scoring path. The former contains a downsample projection layer with parameters $\mathbf{W}_{dn} \in \mathbb{R}^{d \times r}$, a GeLU activation function, and an upsample projection layer with parameters $\mathbf{W}_{up} \in \mathbb{R}^{r \times d}$. The gated scoring path, on the other hand, comprises a scoring weight matrix $\mathbf{W}_s \in \mathbb{R}^{d \times 1}$ and a ReLU activation function. This path is designed to compute a dynamic scaling factor for each token, which regulates the influence of the adaptation process in a data-driven manner. Then these two outputs are multiplied element-wise. Overall, for an input feature $\mathbf{F}_i$, the process of the adapter (AD) can be described as:

$$\text{AD}(\mathbf{F}_i) = \text{ReLU}\left(\mathbf{F}_i \mathbf{W}_s\right) \odot \text{GeLU}\left(\mathbf{F}_i \mathbf{W}_{dn}\right) \mathbf{W}_{up} \quad (3)$$

This two-path design ensures that the adapter module can effectively control the contribution of the adapted features. We use two adapters added to each Transformer layer, parallel to the MHSA and FFN layers as:

$$\hat{\mathbf{F}}_{i-1} = \text{MHSA}(\text{LN}(\mathbf{F}_{i-1})) + \mathbf{F}_{i-1} + \text{AD}(\text{LN}(\mathbf{F}_{i-1})), \quad (4)$$

$$\mathbf{F}_i = \text{FFN}(\text{LN}(\hat{\mathbf{F}}_{i-1})) + \hat{\mathbf{F}}_{i-1} + \text{AD}(\text{LN}(\hat{\mathbf{F}}_{i-1})) \quad (5)$$

3.3. Mixture-of-Geometry-Experts

Although the above adapter enables efficient transfer learning without modifying the core parameters of the pretrained model, its performance still exhibits limitations in cross-category scenarios (Table 4). This is due to the fact that pretraining datasets come from different data domains (e.g., ShapeNet [2] primarily consists of indoor objects), creating a significant gap with our real-world 3D SOT scenarios. Therefore, geometric disparities persist even though they are partially alleviated by leveraging geometric priors. Our solution draws inspiration from MoE [38], which learns multiple data bias views using a set of experts. Although MoE was originally proposed for building large pre-trained models, we adapt it to the context of 3D transfer learning for geometry-aware modeling, referred to as Mixture-of-Geometry-Experts (MoGE), and demonstrate its effectiveness in enhancing category-unified generalization.

The input $\mathbf{Z}_j$ to the jth MoGE layer is constructed by concatenating the temporal token with the search tokens. As shown in Fig. 2(c), the MoGE layer consists of M geometry experts $\{\{\mathbf{E}_j^m\}_{j=1}^L\}_{m=1}^M$, where $\mathbf{E}_j^m$ represents the mth expert at the jth layer and has a same structure as the FFN. The routing algorithm of MoGE determines which experts process inputs. Here we employ the Top-K gating routing mechanisms as our Router $\{\mathbf{R}_j\}_{j=1}^L$ to make decisions using a learnable gate network, which includes an expert embedding $\mathbf{W}_j^R \in \mathbb{R}^{d \times M}$ to transfer features to scores. Specif-

ICCV
#10075

ICCV
#10075

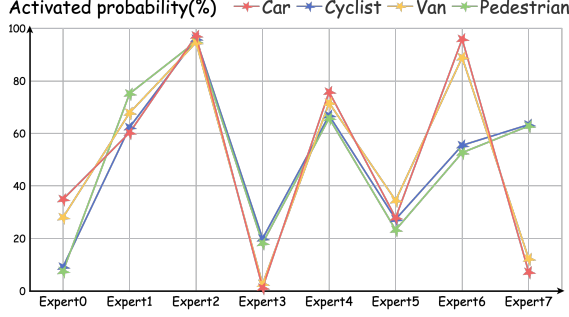ICCV 2025 Submission #10075. CONFIDENTIAL REVIEW COPY. DO NOT DISTRIBUTE.

Figure 3. **Geometry Sensitivity Analysis MoGE.** The distribution of activated experts for different object categories on KITTI. Pedestrian and Cyclist show significantly higher activation in Expert 3 and Expert 7, indicating that these experts excel at handling non-rigid and deformable geometry. In contrast, Van and Car exhibit higher activation in Expert 0 and Expert 6, suggesting that these experts focus on rigid structures. Additionally, some experts (such as Expert 2 and Expert 4) display similar activation trends across all categories, implying that they may have learned general geometric features applicable to various object types.

ically, the MoGE output can be represented as:

$$\mathbf{E}_j(\mathbf{Z}_j) = \sum_{m=1}^{M} \mathbf{R}_j(\mathbf{Z}_j, K)\mathbf{E}_j^m(\mathbf{Z}_j), \tag{6}$$

$$\mathbf{R}_j(\mathbf{Z}_j, K) = \text{Softmax}\left(\text{Top-K}\left(\mathbf{Z}_j\mathbf{W}_j^R, K\right)\right) \tag{7}$$

Only K ($K < M$) experts will be activated, receiving the input and performing adaptive fusion before outputting the result. As shown in the Fig. 3, MoGE demonstrates its design effectiveness by performing adaptive selection based on geometric characteristics rather than relying solely on class labels. We place our MoGE layer after the FFN inside Transformer blocks to avoid disrupting the original structure of the pretrained model.

3.4. Temporal Context Optimization

The initial pretrained model learns representations for static tasks, whereas tracking is inherently a dynamic task. Therefore, we explore additional temporal modeling methods. Inspired by prompt learning [17, 44, 58, 60], we first define a learnable initial temporal token $\mathcal{T}_0 \in \mathbb{R}^{1 \times d}$ that aims to sufficiently interact with all template and search tokens throughout the encoder. In this way, the temporal token absorbs spatiotemporal representations relevant to the current time step.

In detail, for a sequence of T frames, $\mathcal{T}_0$ will be propagated and updated along the time dimension, as illustrated in Fig. 4. At time step t, the input temporal token is updated to $\mathcal{T}_0^t$, which is obtained by combining the learned initial temporal token $\mathcal{T}_0$ with the historical output temporal token $\mathcal{T}_{out}^{t-1}$ from the most recent historical frame's encoder output:

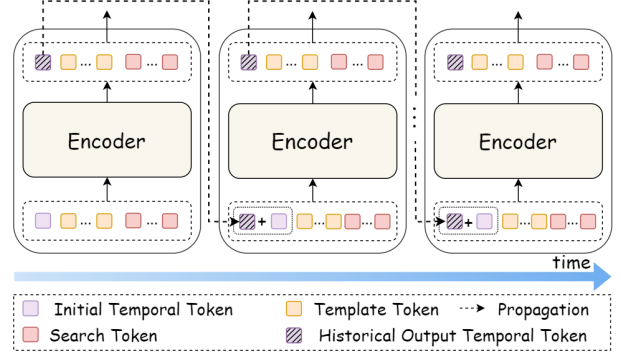$$\mathcal{T}_0^t = \mathcal{T}_0 + \mathcal{T}_{out}^{t-1} \tag{8}$$

Figure 4. **Temporal token propagation**. The temporal token captures key features across different frames and propagates to subsequent frames sequentially.

Note that if $t = 1$, then $\mathcal{T}_0^t = \mathcal{T}_0$. The propagated token $\mathcal{T}_0^t$ is then integrated with current template and search tokens via the encoder. For the training process, $\mathcal{T}_0$ is randomly initialized and continuously updated along with the network. During testing, $\mathcal{T}_0$ is loaded with the trained values. This operation preserves temporal coherence by propagating historical features while avoiding complex computations.

In addition, to address spatiotemporal variations across object categories, we propose a dynamic mask weighting (DMW) mechanism. We first construct masks [57] for the input point clouds, defining a target-oriented mask $\mathcal{M}^t$ where object regions are assigned a value of 0.8 and background regions a value of 0.2, while the search frame uses a uniform mask $\mathcal{M}^s$ initialized to 0.5. We then propose to apply learnable weights $\beta^t \in \mathbb{R}^{N_t \times 1}$, $\beta^s \in \mathbb{R}^{N_s \times 1}$ for template and search regions, which adaptively scale the masks through element-wise multiplication. The adjusted masks are then simply added to the embedded input tokens. Then, the whole process of the dynamic mask weighting module can be expressed as:

$$\tilde{\mathcal{F}}^* = \mathcal{F}^* + \mathcal{M}^* \odot \beta^*, \quad \forall * \in \{t, s\} \tag{9}$$

By jointly optimizing β^* during training, our method not only enhances mask quality through adaptive temporal noise suppression but also dynamically adjusts mask emphasis based on category-specific characteristics without manual hyperparameter tuning.

4. Experiments

4.1. Implementation Details.

We conduct comprehensive experiments using three widely-used datasets: KITTI [9], NuScenes [1], and Waymo Open Dataset (WOD) [39]. We sample clips from each sequence to form training samples. The length of each clip is set to 3 frames. Specifically, for the template and search regions, we set the number of input points as N_t=128

ICCV
#10075

ICCV
#10075

ICCV 2025 Submission #10075. CONFIDENTIAL REVIEW COPY. DO NOT DISTRIBUTE.

Table 1. Comparison with SOTA methods on KITTI. ▮ and ▮ refer to category-specific methods and category-unified methods, respectively. **Bold** and <u>underline</u> denote current and previous best performance. "TP" denotes Tunable Parameters.

Method	FPS	TP(M)	Car [6,424]	Pedestrian [6,088]	Van [1,248]	Cyclist [308]	Mean [14,068]
P2B [34]	40	1.34	56.2 / 72.8	28.7 / 49.6	40.8 / 48.4	32.1 / 44.7	42.4 / 60.0
V2B [14]	37	1.36	70.5 / 81.3	48.3 / 73.5	50.1 / 58.0	40.8 / 49.7	58.4 / 75.2
PTTR [59]	50	2.27	65.2 / 77.4	50.9 / 81.6	52.5 / 61.8	65.1 / 90.5	57.9 / 78.2
STNet [15]	35	1.66	72.1 / 84.0	49.9 / 77.2	58.0 / 70.6	73.5 / 93.7	61.3 / 80.1
GLT-T [27]	30	2.60	68.2 / 82.1	52.4 / 78.8	52.6 / 62.9	68.9 / 92.1	60.1 / 79.3
CXTrack [47]	29	18.30	69.1 / 81.6	67.0 / 91.5	60.0 / 71.8	74.2 / <u>94.3</u>	67.5 / 85.3
M^{2}Track [57]	57	2.24	65.5 / 80.8	61.5 / 88.2	53.8 / 70.7	73.2 / 93.5	62.9 / 83.4
SyncTrack [26]	45	1.47	73.3 / <u>85.0</u>	54.7 / 80.5	60.3 / 70.0	73.1 / 93.8	64.1 / 81.9
MBPTrack [48]	50	7.38	<u>73.4</u> / 84.8	<u>68.6</u> / <u>93.9</u>	**61.3** / 72.7	<u>76.7</u> / <u>94.3</u>	<u>70.3</u> / 87.9
PTTR++ [25]	43	-	<u>73.4</u> / 84.5	55.2 / 84.7	55.1 / 62.2	71.6 / 92.8	63.9 / 82.8
M3SOT [22]	38	-	**75.9 / 87.4**	66.6 / 92.5	59.4 / <u>74.7</u>	70.3 / 93.4	<u>70.3</u> / **88.6**
StreamTrack [24]	41	-	72.6 / 83.7	**70.5 / 94.7**	<u>61.0</u> / **76.9**	**78.1 / 94.6**	**70.8** / <u>88.1</u>
STNet [15]	35	1.66	<u>69.5</u> / <u>80.5</u>	43.8 / 67.4	58.1 / 68.2	73.8 / <u>94.0</u>	57.5 / 74.0
CXTrack [47]	29	18.30	60.2 / 72.6	54.6 / 81.6	57.6 / 70.0	44.4 / 57.0	57.2 / 75.9
M^{2}Track [57]	57	2.24	62.9 / 77.0	57.4 / 84.4	66.2 / <u>80.7</u>	76.0 / 93.7	61.1 / 80.9
MBPTrack [48]	50	7.38	62.3 / 72.1	50.2 / 80.9	<u>66.6</u> / 78.2	71.8 / 92.2	56.1 / 74.9
SiamCUT [28]	36	2.06	58.1 / 73.9	48.2 / 76.2	63.1 / 74.9	36.7 / 47.4	54.0 / 74.6
MoCUT [28]	48	2.34	67.6 / <u>80.5</u>	**63.3 / 90.0**	64.5 / 78.8	<u>76.7</u> / 94.2	<u>65.8</u> / <u>85.0</u>
TrackAny3D	28	5.30	**73.4 / 85.2**	<u>59.6</u> / <u>85.6</u>	**70.0 / 82.8**	74.7 / <u>94.0</u>	**67.1 / 85.4**

and N_s=128 after farthest point sampling, respectively. Our model is built on the pretrained RECON [35] with its original parameters frozen. The bottleneck dimension r of our adapters is set to 72. For MoGE module, we use M=8 experts and select K=4. The FFN for MoGE, the intermediate bottleneck dimension is set to 1/8 of the Transformer dimension. The MoGE layer is placed at every even-number layer. Inference for our model was conducted on a single NVIDIA RTX3090 GPU. More details are provided in the appendix.

4.2. Comparison with State-of-the-art Methods

In this section, we conduct extensive comparisons on three datasets, categorizing all methods into two experimental settings: category-specific and category-unified . In the category-unified setting, to facilitate fair comparisons, we reproduce the results of several SOTA methods, including STNet [15], M^{2}Track [57], CXTrack [47], and MBP-Track [48], using their officially open-sourced code.
Results on KITTI. In Table 1, we compare our method against SOTA methods on KITTI. Recent category-specific models such as STNet, MBPTrack, and StreamTrack have shown steady improvements in accuracy, indicating rapid advancements in single-category model design. However, we observe that when these models are trained under the category-unified setting, their performance drops significantly. For instance, STNet, CXTrack, and MBPTrack experience performance declines of 3.8%, 10.3%, and 14.2%, respectively. In contrast, our method is comparable to those of category-specific methods and achieves the best results

when trained on all categories together, reaching a success rate of 67.1%, surpassing all other methods in this setting, including the latest MoCUT, which focuses on the same problem. Specifically, our method outperforms MoCUT by 1.3% overall and by 6.0% in the Car category. These results demonstrate the effectiveness of our approach, showcasing its ability to generalize well across multiple categories while maintaining high performance.
Results on NuScenes. Table 2 presents the comparison results on NuScenes. Our method achieves a consistent and significant performance gain compared to previous SOTA methods, such as MoCUT, under the category-unified setting, while remaining competitive compared to category-specific methods. Additionally, we observe that in this dataset, due to the large amount of data for common categories like Cars and Pedestrians, the performance of less frequent categories (including Trailers, Trucks, Buses) is significantly improved, even surpassing methods like MBP-Track, which are specifically trained for certain categories. Similar to the trend observed on KITTI, competing methods still exhibit a substantial performance drop between their category-unified and category-specific results; for example, MBPTrack shows a noticeable degradation in performance of 6.35%. In contrast, our TrackAny3D achieves the best results under the category-unified setting, with results on individual classes such as Bus even surpassing all single-category learning models. This highlights the superiority of our method in addressing geometric disparity issues.
Results on WOD. In Table 3, we perform direct inference on the Vehicle of WOD using our model trained on KITTI, thereby validating the generalization capability of our ap-

ICCV
#10075

ICCV
#10075

ICCV 2025 Submission #10075. CONFIDENTIAL REVIEW COPY. DO NOT DISTRIBUTE.

Table 2. Comparison with SOTA on NuScenes.

Method	Car [64,159]	Pedestrian [33,227]	Truck [13,587]	Trailer [3,352]	Bus [2,953]	Mean [117,278]
SC3D [10]	22.31 / 21.93	11.29 / 12.65	30.67/ 27.73	35.28 / 28.12	29.35 / 24.08	20.70 / 22.20
P2B [34]	38.81 / 43.18	28.39 / 52.24	42.95 / 41.59	48.96 / 40.05	32.95 / 27.41	36.48 / 45.08
PTT [37]	41.22 / 45.26	19.33 / 32.03	50.23 / 48.56	51.70 / 46.50	39.40 / 36.70	36.33 / 41.72
PTTR [59]	51.89 / 58.61	29.90 / 45.09	45.30 / 44.74	45.87 / 38.36	43.14 / 37.74	44.50 / 52.07
GLT-T [27]	48.52 / 54.29	31.74 / 56.49	52.74 / 51.43	57.60 / 52.01	44.55 / 40.69	44.42 / 54.33
CXTrack [47]	42.64 / 47.96	32.75 / 59.33	40.31 / 38.73	51.48 / 41.36	38.76 / 30.67	39.72 / 49.51
M^{2}Track [57]	55.85 / 65.09	32.10 / 60.72	57.36 / 59.54	57.61 / 58.26	51.39 / 51.44	49.32 / 62.73
SeqTrack3D[20]	**62.55 / 71.46**	39.94 / 68.57	60.97 / 63.04	**68.37** / 61.76	54.33 / 53.52	55.92 / 68.94
MBPTrack[48]	62.47 / 70.41	**45.32 / 74.03**	62.18 / 63.31	65.14 / 61.33	55.41 / 51.76	**57.48 / 69.88**
PTTR++[25]	59.96 / 66.73	32.49 / 50.50	59.85 / 61.20	54.51 / 50.28	53.98 / 51.22	51.86 / 60.63
SCVTrack[53]	58.90 / 67.70	34.50 / 61.50	60.60 / 61.40	59.50 / 60.10	54.30 / 53.60	52.10 / 64.70
StreamTrack[24]	62.05 / 70.81	38.43 / 68.58	**64.67 / 66.60**	66.67 / **64.27**	**60.66 / 59.74**	55.75 / 69.22
CXTrack [47]	43.43 / 47.42	34.19 / 61.53	48.24 / 44.84	54.52 / 42.41	43.21 / 35.41	41.69 / 50.67
M^{2}Track [57]	50.76 / 57.62	28.35 / 53.53	55.97 / 57.39	47.87 / 42.09	51.65 / 49.10	44.98 / 55.76
MBPTrack [48]	55.77 / 61.46	37.62 / 65.38	58.99 / 57.23	58.11 / 46.34	58.37 / 54.20	51.13 / 61.47
SiamCUT [28]	40.96 / 44.91	31.42 / 53.80	53.91 / 52.65	63.29 / 58.21	41.03 / 38.01	40.41 / 48.54
MoCUT [28]	57.32 / 66.01	33.47 / 63.12	61.75 / **64.38**	60.90 / **61.84**	57.39 / 56.07	51.19 / 64.63
TrackAny3D	59.30 / 66.46	**40.37 / 68.70**	**62.70** / 62.80	**66.12** / 59.20	**61.01 / 58.02**	**54.57 / 66.25**

proach. For the category-specific methods, following their work, we use their KITTI Car models. The results indicate that TrackAny3D demonstrates strong competitiveness and achieves the best tracking performance with 64%, outperforming all other methods, including those designed for category-specific tasks. Specifically, compared to MoCUT, our method improves performance by 2.1%, and compared to MBPTrack under the category-unified setting, we achieve a lead of 5.8%. These results highlight that TrackAny3D, under the new paradigm of pretrained model transfer, possesses excellent generalization capabilities, effectively addressing cross-dataset and cross-category challenges.

4.3. Ablation Studies

In this section, we conduct extensive ablation studies on TrackAny3D using the KITTI dataset under the category-unified setting, analyzing the impact of the model's core components and settings. Additional ablation results are provided in the supplementary material.

Model components. Table 4 presents the ablation study of the components in TrackAny3D. The experimental results indicate that when RECON is fully finetuned, the average metric of the model is limited. However, when we freeze the parameters of RECON and introduce our two-path adapter for transfer learning, the average metric improves significantly. Furthermore, by progressively incorporating our MoGE, learnable temporal token, and dynamic weighted mask, the performance continues to increase, demonstrating the effectiveness of each component.

Position of MoGE and Adapters. We analyze the impact of insertion positions for adapters and MoGE in Table 5. For adapters, we observe that performance improves with

Table 3. Comparison with SOTA on WOD.

Method	Vehicle[185731]			
	Easy	Medium	Hard	Mean
P2B[34]	57.1 / 65.4	52.0 / 60.7	47.9 / 58.5	52.6 / 61.7
BAT[56]	61.0 / 68.3	53.3 / 60.9	48.9 / 57.8	54.7 / 62.7
V2B[14]	64.5 / 71.5	55.1 / 63.2	52.0 / 62.0	57.6 / 65.9
STNet[15]	65.9 / 72.7	57.5 / 66.0	54.6 / 64.7	59.7 / 68.0
TAT[18]	66.0 / 72.6	56.6 / 64.2	52.9 / 62.5	58.9 / 66.7
CXTrack[47]	63.9 / 71.1	54.2 / 62.7	52.1 / 63.7	57.1 / 66.1
M^{2}Track[57]	68.1 / 75.3	**58.6** / 66.6	55.4 / 64.9	61.1 / 69.3
MBPTrack[48]	**68.5 / 77.1**	58.4 / **68.1**	**57.6 / 69.7**	**61.9 / 71.9**
SiamDisst[45]	-	-	-	61.2 / 70.5
MemDisst[45]	-	-	-	**61.9 / 71.9**
CXTrack[47]	60.7 / 67.6	50.5 / 57.7	46.0 / 55.8	52.8 / 60.7
STNet[15]	67.5 / 75.2	59.2 / 68.2	55.6 / 66.4	61.1 / 70.2
MBPTrack[48]	65.6 / 74.4	55.4 / 64.6	52.5 / 63.2	58.2 / 67.7
SiamCUT[28]	58.3 / 66.0	50.8 / 60.8	49.2 / 59.1	53.0 / 62.2
MoCUT[28]	68.3 / 75.0	59.4 / 66.9	57.1 / 66.3	61.9 / 69.7
TrackAny3D	**72.6 / 80.2**	**60.5 / 69.6**	**57.5 / 68.9**	**64.0 / 73.3**

the number of layers they are added to; thus, we ultimately incorporate them into all layers. In contrast, for MoGE, adding it to all layers results in an obvious performance drop, which we attribute to overfitting caused by excessive parameters. The best performance is achieved when MoGE is applied to half of the layers, and this is our final setting.

Length of Temporal Propagation. We investigate the impact of the number of sampled frames for training temporal tokens in Table 6. Note that setting the number of propagation frames to 2 corresponds to not using temporal tokens. It is observed that using 3 frames yields the best performance. However, further increasing the sequence

7

ICCV
#10075

ICCV
#10075

ICCV 2025 Submission #10075. CONFIDENTIAL REVIEW COPY. DO NOT DISTRIBUTE.

Table 4. Ablations of different model components. "☐" represent the final setting of our TrackAny3D. "FF" denotes Full Fine-tuning. "AD" denotes Adapter Tuning. "TT" denotes Temporal Token.

FF	AD	MoGE	TT	DMW	Car	Pedestrian	Van	Cyclist	Mean
✓	✗	✗	✗	✗	69.8 / 83.6	53.6 / 81.6	62.2 / 73.6	65.8 / 90.8	62.0 / 82.0
✓	✗	✓	✓	✓	72.5 / 84.0	57.9 / 82.0	70.8 / 82.9	72.8 / 93.7	66.0 / 83.3
✗	✓	✗	✗	✗	71.8 / 85.6	55.8 / 83.9	69.8 / 82.7	74.1 / 94.1	64.8 / 84.8
✗	✓	✓	✗	✗	72.5 / 84.8	56.3 / 83.5	71.2 / 83.2	67.4 / 92.1	65.3 / 84.3
✗	✓	✓	✓	✗	72.5 / 84.4	58.1 / 85.7	71.9 / 83.8	72.4 / 89.7	66.2 / 85.0
✗	✓	✓	✓	✓	73.4 / 85.2	59.6 / 85.6	70.0 / 82.8	74.7 / 94.0	67.1 / 85.4

Table 5. Ablations for positions of adapters(AD) and MoGE.

	Position	Car	Pedestrian	Van	Cyclist	Mean
AD	all layers	73.4/85.2	59.6/85.6	70.0/82.8	74.7/94.0	67.1/85.4
	even layers	69.0/80.0	60.0/86.0	72.3/84.4	73.4/93.1	65.5/83.3
	last layer	70.5/80.7	51.2/73.9	70.3/81.5	71.5/92.9	62.2/78.1
MoGE	all layers	71.8/83.5	53.6/81.7	69.5/81.3	72.5/93.5	63.7/82.7
	even layers	73.4/85.2	59.6/85.6	70.0/82.8	74.7/94.0	67.1/85.4
	last layer	70.2/81.8	50.3/78.3	71.5/82.8	71.7/92.6	61.7/80.6

Table 6. Ablations for length of temporal propagation.

Length	Car	Pedestrain	Van	Cyclist	Mean
2	69.5/80.9	56.1/82.5	73.0/84.5	66.9/90.8	64.0/82.1
3	73.4/85.2	59.6/85.6	70.0/82.8	74.7/94.0	67.1/85.4
5	71.3/82.1	54.8/78.6	69.9/82.3	74.6/94.0	64.1/80.9

Table 7. Ablations for dynamic mask weight.

Methods	Car	Pedestrian	Van	Cyclist	Mean
w/o β	71.5/84.4	58.1/85.7	71.9/83.8	64.4/89.7	65.6/85.0
with β	73.4/85.2	59.6/85.6	70.0/82.8	74.7/94.0	67.1/85.4
LM	73.8/85.3	56.4/83.9	72.6/84.6	72.7/93.0	66.1/84.8

length does not lead to performance gains, suggesting that overly long search video clips impose a learning burden on the model. Therefore, it is important to choose an appropriate length for the search video clip.

Influence of Dynamic Mask Weighting. In Table 7, we explore the effects of masks under different settings. Here, "w/o β" indicates directly using the mask, "with β" refers to our dynamic weighting mask, and "LM" means initializing the mask as a learnable parameter. The results show that incorporating learnable components into the mask is beneficial. Moreover, multiplying the mask by an additional learnable β achieves better performance compared to directly making the mask fully learnable.

4.4. Visualization Analysis

In Fig. 5, we present the qualitative results on KITTI under category-unified settings. It is evident that even when dealing with scenarios where object surface points are extremely sparse, such as those involving Car and Van, our method still demonstrates significant advantages. Our TrackAny3D closely aligns with the ground truth, whereas
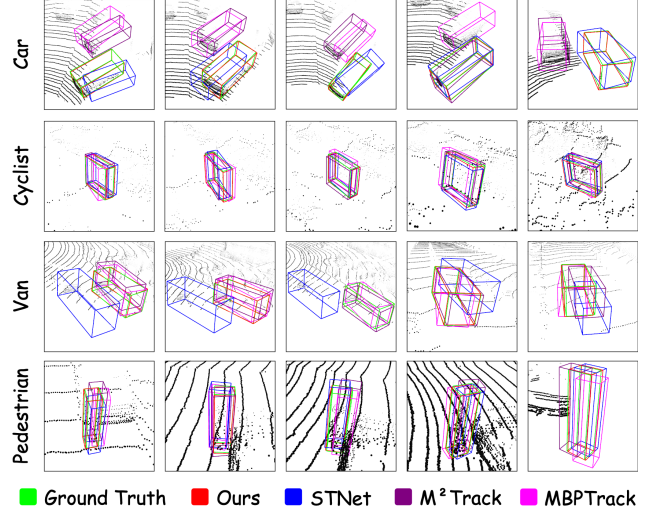

Figure 5. **Qualitative Visualization.**

🟩 Ground Truth 🟥 Ours 🟦 STNet 🟪 M²Track 🟪 MBPTrack

other methods exhibit noticeable deviations or lose track of the target. Additionally, in the tracking of Pedestrian and Cyclist, our method also shows higher accuracy and stability. Even in scenes with dense crowds or background interference, our method can continuously provide reliable target localization without being easily affected by surrounding environmental disturbances. These qualitative results not only validate the superiority of our method in handling geometric disparities but also demonstrate its effectiveness and reliability in complex environments.

5. Conclusion

In this paper, we present TrackAny3D , the first framework to bridge large-scale pretrained point cloud models with category-agnostic 3D SOT. By integrating parameter-efficient adapters and a geometry-aware expert architecture, TrackAny3D effectively transfers geometric priors from pretraining while adaptively resolving cross-category disparities. The proposed temporal context optimization further enhances robustness to temporal changes and feature calibration. Extensive experiments demonstrate that TrackAny3D achieves SOTA performance under category-unified settings while remaining competitive with category-specific methods, showcasing strong generalization across diverse real-world scenarios.

ICCV
#10075

ICCV
#10075

ICCV 2025 Submission #10075. CONFIDENTIAL REVIEW COPY. DO NOT DISTRIBUTE.

References

[1] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020. 2, 5

[2] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 2, 4

[3] Ying Cui, Dongyan Guo, Yanyan Shao, Zhenhua Wang, Chunhua Shen, Liyan Zhang, and Shengyong Chen. Joint classification and regression for visual tracking with fully convolutional siamese networks. *International Journal of Computer Vision*, pages 1–17, 2022. 1

[4] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017. 2

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186, 2019. 2

[6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint*, 2020. 3

[7] Zheng Fang, Sifan Zhou, Yubo Cui, and Sebastian Scherer. 3d-siamrpn: An end-to-end learning method for real-time 3d single object tracking using raw point cloud. *IEEE Sensors Journal*, 21(4):4995–5011, 2020. 2

[8] Peng Gao, Shijie Geng, Renrui Zhang, Teli Ma, Rongyao Fang, Yongfeng Zhang, Hongsheng Li, and Yu Qiao. Clip-adapter: Better vision-language models with feature adapters. *arXiv preprint arXiv:2110.04544*, 2021. 2, 3

[9] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3354–3361. IEEE, 2012. 2, 5

[10] Silvio Giancola, Jesus Zarzar, and Bernard Ghanem. Leveraging shape completion for 3d siamese tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1359–1368, 2019. 1, 2, 7

[11] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International conference on machine learning*, pages 2790–2799. PMLR, 2019. 2, 3

[12] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022. 3

[13] Tianyu Huang, Bowen Dong, Yunhan Yang, Xiaoshui Huang, Rynson WH Lau, Wanli Ouyang, and Wangmeng Zuo. Clip2point: Transfer clip to point cloud classification with image-depth pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22157–22167, 2023. 3

[14] Le Hui, Lingpeng Wang, Mingmei Cheng, Jin Xie, and Jian Yang. 3d siamese voxel-to-bev tracker for sparse point clouds. *Advances in Neural Information Processing Systems*, 34:28714–28727, 2021. 2, 6, 7

[15] Le Hui, Lingpeng Wang, Linghua Tang, Kaihao Lan, Jin Xie, and Jian Yang. 3d siamese transformer network for single object tracking on point clouds. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part II*, pages 293–310. Springer, 2022. 1, 2, 6, 7

[16] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *ECCV*, 2022. 2, 3

[17] Muhammad Uzair Khattak, Hanoona Rasheed, Muhammad Maaz, Salman Khan, and Fahad Shahbaz Khan. Maple: Multi-modal prompt learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19113–19122, 2023. 3, 5

[18] Kaihao Lan, Haobo Jiang, and Jin Xie. Temporal-aware siamese tracker: Integrate temporal context for 3d object tracking. In *Proceedings of the Asian Conference on Computer Vision*, pages 399–414, 2022. 7

[19] Xin Li, Yuqing Huang, Zhenyu He, Yaowei Wang, Huchuan Lu, and Ming-Hsuan Yang. Citetracker: Correlating image and text for visual tracking. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9974–9983, 2023. 1, 2

[20] Yu Lin, Zhiheng Li, Yubo Cui, and Zheng Fang. Seqtrack3d: Exploring sequence information for robust 3d point cloud tracking. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6959–6965. IEEE, 2024. 2, 7

[21] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023. 2, 3

[22] Jiaming Liu, Yue Wu, Maoguo Gong, Qiguang Miao, Wenping Ma, Cai Xu, and Can Qin. M3sot: multi-frame, multi-field, multi-space 3d single object tracking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 3630–3638, 2024. 2, 6

[23] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019. 2

ICCV
#10075

ICCV
#10075

ICCV 2025 Submission #10075. CONFIDENTIAL REVIEW COPY. DO NOT DISTRIBUTE.

[24] Zhipeng Luo, Gongjie Zhang, Changqing Zhou, Zhonghua Wu, Qingyi Tao, Lewei Lu, and Shijian Lu. Modeling continuous motion for 3d point cloud object tracking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 4026–4034, 2024. 1, 2, 6, 7

[25] Zhipeng Luo, Changqing Zhou, Liang Pan, Gongjie Zhang, Tianrui Liu, Yueru Luo, Haiyu Zhao, Ziwei Liu, and Shijian Lu. Exploring point-bev fusion for 3d point cloud object tracking with transformer. *IEEE transactions on pattern analysis and machine intelligence*, 2024. 6, 7

[26] Teli Ma, Mengmeng Wang, Jimin Xiao, Huifeng Wu, and Yong Liu. Synchronize feature extracting and matching: A single branch framework for 3d object tracking. *arXiv preprint arXiv:2308.12549*, 2023. 1, 2, 6

[27] Jiahao Nie, Zhiwei He, Yuxiang Yang, Mingyu Gao, and Jing Zhang. Glt-t: Global-local transformer voting for 3d single object tracking in point clouds. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1957–1965, 2023. 6, 7

[28] Jiahao Nie, Zhiwei He, Xudong Lv, Xueyi Zhou, Dong-Kyu Chae, and Fei Xie. Towards category unification of 3d single object tracking on point clouds. *arXiv preprint arXiv:2401.11204*, 2024. 2, 6, 7

[29] Jiahao Nie, Fei Xie, Sifan Zhou, Xueyi Zhou, Dong-Kyu Chae, and Zhiwei He. P2p: Part-to-part motion cues guide a strong tracking framework for lidar point clouds. *arXiv preprint arXiv:2407.05238*, 2024. 1, 2

[30] Junting Pan, Ziyi Lin, Xiatian Zhu, Jing Shao, and Hongsheng Li. St-adapter: Parameter-efficient image-to-video transfer learning. *Advances in Neural Information Processing Systems*, 35:26462–26477, 2022. 2, 3

[31] Yatian Pang, Wenxiao Wang, Francis EH Tay, Wei Liu, Yonghong Tian, and Li Yuan. Masked autoencoders for point cloud self-supervised learning. In *European conference on computer vision*, pages 604–621. Springer, 2022. 2, 3

[32] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. 3

[33] Charles R Qi, Or Litany, Kaiming He, and Leonidas J Guibas. Deep hough voting for 3d object detection in point clouds. In *proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9277–9286, 2019. 2

[34] Haozhe Qi, Chen Feng, Zhiguo Cao, Feng Zhao, and Yang Xiao. P2b: Point-to-box network for 3d object tracking in point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6329–6338, 2020. 1, 2, 6, 7

[35] Zekun Qi, Runpei Dong, Guofan Fan, Zheng Ge, Xiangyu Zhang, Kaisheng Ma, and Li Yi. Contrast with reconstruct: Contrastive 3d representation learning guided by generative pretraining. In *International Conference on Machine Learning*, pages 28223–28243. PMLR, 2023. 3, 6

[36] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, pages 8748–8763. PMLR, 2021. 2, 3

[37] Jiayao Shan, Sifan Zhou, Zheng Fang, and Yubo Cui. Ptt: Point-track-transformer module for 3d single object tracking in point clouds. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1310–1316. IEEE, 2021. 7

[38] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017. 4

[39] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2446–2454, 2020. 2, 5

[40] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. 2, 3

[41] Mengmeng Wang, Teli Ma, Xingxing Zuo, Jiajun Lv, and Yong Liu. Correlation pyramid network for 3d single object tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3216–3225, 2023. 1, 2

[42] Mengmeng Wang, Jiazheng Xing, Jianbiao Mei, Yong Liu, and Yunliang Jiang. Actionclip: Adapting language-image pretrained models for video action recognition. *IEEE Transactions on Neural Networks and Learning Systems*, 2023. 2, 3

[43] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics (tog)*, 38(5):1–12, 2019. 3

[44] Syed Talal Wasim, Muzammal Naseer, Salman Khan, Fahad Shahbaz Khan, and Mubarak Shah. Vita-clip: Video and text adaptive clip via multimodal prompting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23034–23044, 2023. 5

[45] Qiangqiang Wu, Yan Xia, Jia Wan, and Antoni B Chan. Boosting 3d single object tracking with 2d matching distillation and 3d pre-training. In *European Conference on Computer Vision*, pages 270–288. Springer, 2024. 3, 7

[46] Saining Xie, Jiatao Gu, Demi Guo, Charles R Qi, Leonidas Guibas, and Or Litany. Pointcontrast: Unsupervised pre-training for 3d point cloud understanding. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*, pages 574–591. Springer, 2020. 2

[47] Tian-Xing Xu, Yuan-Chen Guo, Yu-Kun Lai, and Song-Hai Zhang. Cxtrack: Improving 3d point cloud tracking with contextual information. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1084–1093, 2023. 1, 2, 6, 7

ICCV
#10075

ICCV 2025 Submission #10075. CONFIDENTIAL REVIEW COPY. DO NOT DISTRIBUTE.

ICCV
#10075

[48] Tian-Xing Xu, Yuan-Chen Guo, Yu-Kun Lai, and Song-Hai Zhang. Mbptrack: Improving 3d point cloud tracking with memory networks and box priors. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9911–9920, 2023. 1, 2, 3, 6, 7

[49] Yuxiang Yang, Yingqi Deng, Jing Zhang, Hongjie Gu, and Zhekang Dong. Siammo: Siamese motion-centric 3d object tracking. *arXiv preprint arXiv:2408.01688*, 2024. 2

[50] Botao Ye, Hong Chang, Bingpeng Ma, Shiguang Shan, and Xilin Chen. Joint feature learning and relation modeling for tracking: A one-stream framework. In *European conference on computer vision*, pages 341–357. Springer, 2022. 1, 3

[51] Xumin Yu, Lulu Tang, Yongming Rao, Tiejun Huang, Jie Zhou, and Jiwen Lu. Point-bert: Pre-training 3d point cloud transformers with masked point modeling. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 19313–19322, 2022. 2, 3

[52] Yaohua Zha, Jinpeng Wang, Tao Dai, Bin Chen, Zhi Wang, and Shu-Tao Xia. Instance-aware dynamic prompt tuning for pre-trained point cloud models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14161–14170, 2023. 3

[53] Jingwen Zhang, Zikun Zhou, Guangming Lu, Jiandong Tian, and Wenjie Pei. Robust 3d tracking with quality-aware shape completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 7160–7168, 2024. 2, 7

[54] Renrui Zhang, Rongyao Fang, Peng Gao, Wei Zhang, Kunchang Li, Jifeng Dai, Yu Qiao, and Hongsheng Li. Tip-adapter: Training-free clip-adapter for better vision-language modeling. *arXiv preprint arXiv:2111.03930*, 2021. 3

[55] Renrui Zhang, Ziyu Guo, Wei Zhang, Kunchang Li, Xupeng Miao, Bin Cui, Yu Qiao, Peng Gao, and Hongsheng Li. Pointclip: Point cloud understanding by clip. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8552–8562, 2022. 2, 3

[56] Chaoda Zheng, Xu Yan, Jiantao Gao, Weibing Zhao, Wei Zhang, Zhen Li, and Shuguang Cui. Box-aware feature enhancement for single object tracking on point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13199–13208, 2021. 2, 7

[57] Chaoda Zheng, Xu Yan, Haiming Zhang, Baoyuan Wang, Shenghui Cheng, Shuguang Cui, and Zhen Li. Beyond 3d siamese tracking: A motion-centric paradigm for 3d single object tracking in point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8111–8120, 2022. 2, 3, 5, 6, 7

[58] Yaozong Zheng, Bineng Zhong, Qihua Liang, Zhiyi Mo, Shengping Zhang, and Xianxian Li. Odtrack: Online dense temporal token learning for visual tracking. In *Proceedings of the AAAI conference on artificial intelligence*, pages 7588–7596, 2024. 5

[59] Changqing Zhou, Zhipeng Luo, Yueru Luo, Tianrui Liu, Liang Pan, Zhongang Cai, Haiyu Zhao, and Shijian Lu. Pttr: Relational 3d point cloud object tracking with transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8531–8540, 2022. 6, 7

[60] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *IJCV*, 2022. 2, 3, 5

[61] Xin Zhou, Dingkang Liang, Wei Xu, Xingkui Zhu, Yihan Xu, Zhikang Zou, and Xiang Bai. Dynamic adapter meets prompt tuning: Parameter-efficient transfer learning for point cloud analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14707–14717, 2024. 3, 4